This project shows how to design a Digital Read Out (DRO) (used in the Mills/Lathe machines) using a PSoC 6 BLE device. Mills and lathes are economical which do not have DRO installed when purchased. A DRO is simply a digital reading of the scale along the axis in the mill or lathe. Usually, it is the X-, Y- and Z- axis for a milling machine, refer to Figure 1, and X- and Z- axis for lathes, refer to Figure 2. Linear encoders which output the position of the scale can be fixed on the axes of these machines and can be connected to a DRO as shown in Figure 3 and Figure 4. This PSoC 6 BLE DRO measures the position of all the three axes and sends the data over BLE to an Android App.

Figure 1. The axes for a lathe machine

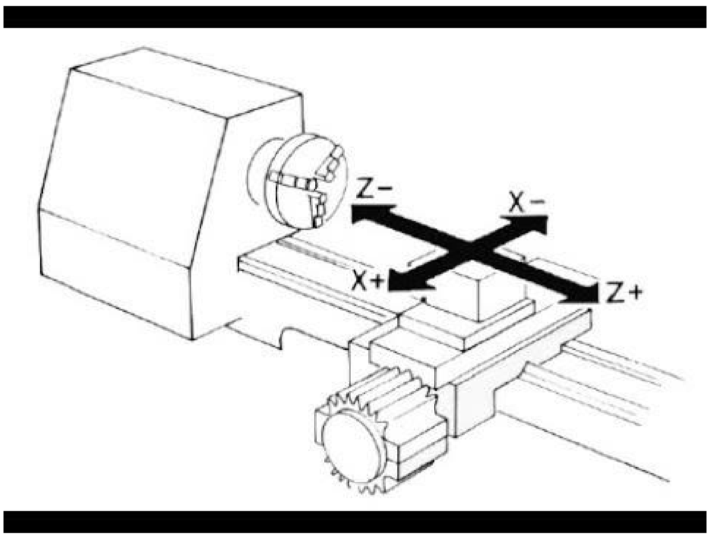


Figure 2. The axes for a milling machine

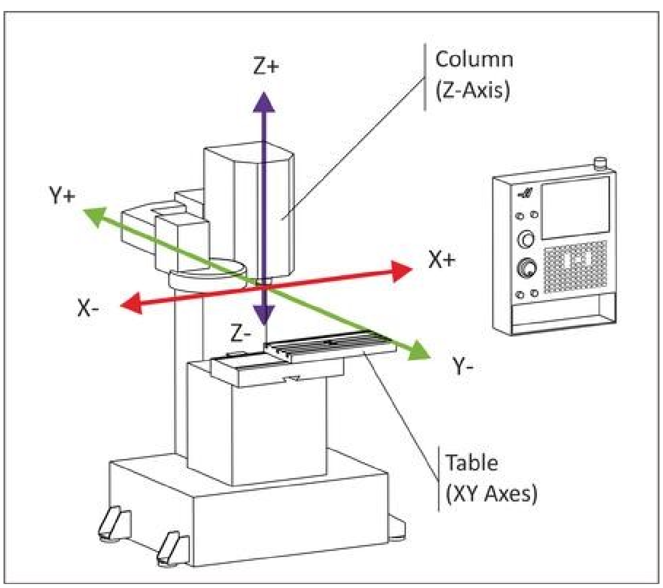


Figure 3. A simple 3-axes Digital Readout

Figure 4. A linear encoder and a digital readout kit

The Linear encoders produce a quadrature digital waveform that can be processed by the PSoC 6's Quadrature decoder component. PSoC 6 can accommodate three quadrature decoders and satisfies this application. PSoC 6 measures the position of the encoders on the corresponding axis and communicates over BLE to an Android App which displays the position information. In the App, the encoder readings can be displayed as X axis, Y axis and Z axis readings as shown in Figure 10. This PSoC 6 application will accommodate at least 100mm on each axis at a resolution of 1 micron. This translates to 100,000 counts which implies a 17-bit counter is required.

Requirements:

Tool: PSoC Creator™ 4.3, Serial Terminal application (like TeraTerm)
Programming Language: C (Arm GCC 5.4.1)
Associated Parts: All PSoC 6 MCUs with BLE Connectivity
Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Design:

In the PSoC 6 BLE DRO Project, quadrature decoder components are used to detect the direction of Linear Encoders in all three axes. The TCPWM block is configured as a quadrature decoder for this purpose. The counter of the quadrature decoder is initialized with a value of zero. The counter value changes (increases/ decreases) are based on the quadrature input signal. A positive edge on phiA increments the counter when phiB is 0 and decrements the counter when phiB is 1. Therefore, if phiA leads phiB, the count value of the quadrature decoder increases; if phiB leads phiA, the count value decreases. These count values will be sent to Android BLE App.

The LUT component is used to configure the Glitch filter for the digital inputs.

BLE Component is configured in the GATT Server role with a custom BLE service to demonstrate the PSoC 6 BLE DRO functionality. The GATT Server uses the GATT notification on three custom Characteristics to send data of all three counter values from the quadrature decoder to the GATT client device (Android App) continuously. The timer component is configured with a 200ms period to send the notification data every 200ms to the BLE App.

In the PSoC 6 BLE DRO App, after the connection to the PSoC 6 device, the X axis, Y axis, and Z axis values will be displayed. The App also has the **SET MACHINE PARAMETERS** for setting the full-scale length and full-scale count of the 3 axes that are needed for the calculation, and **WORKSPACE ZERO** to send the reset command to the PSoC 6 device which resets the axis values to zero and starts counting from the beginning.

Figure (1), (2), (3), and (4) show the project schematics.

Figure 5: PSoC6 BLE DRO Schematic

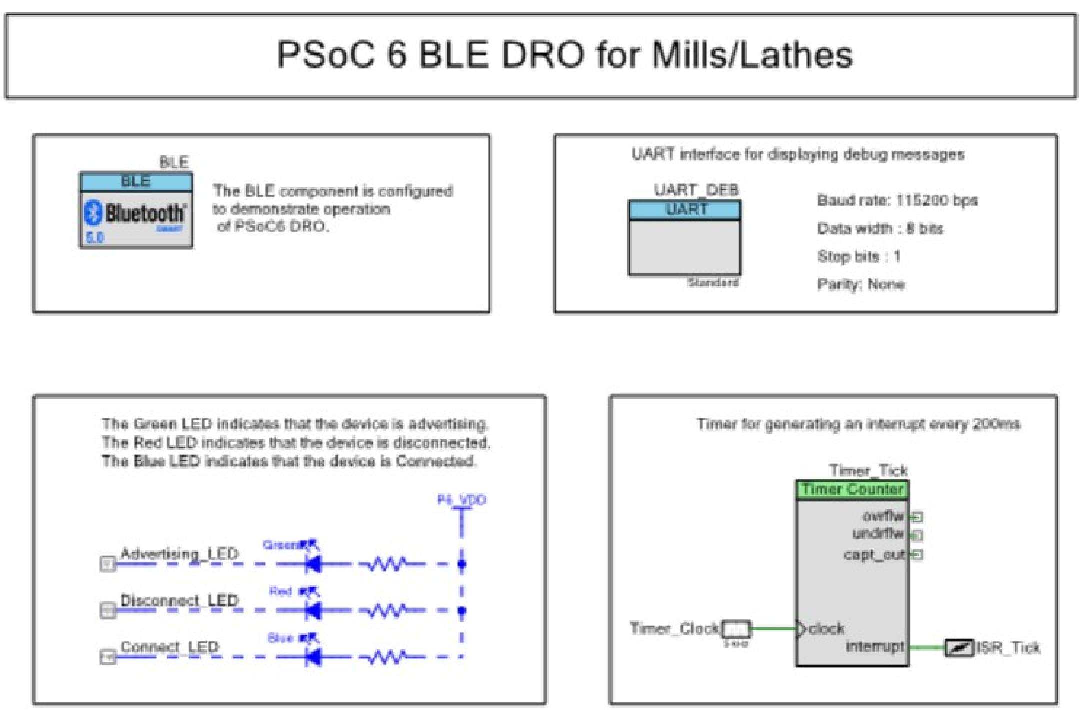


Figure 6: X-Axis Quad Decoder Counter

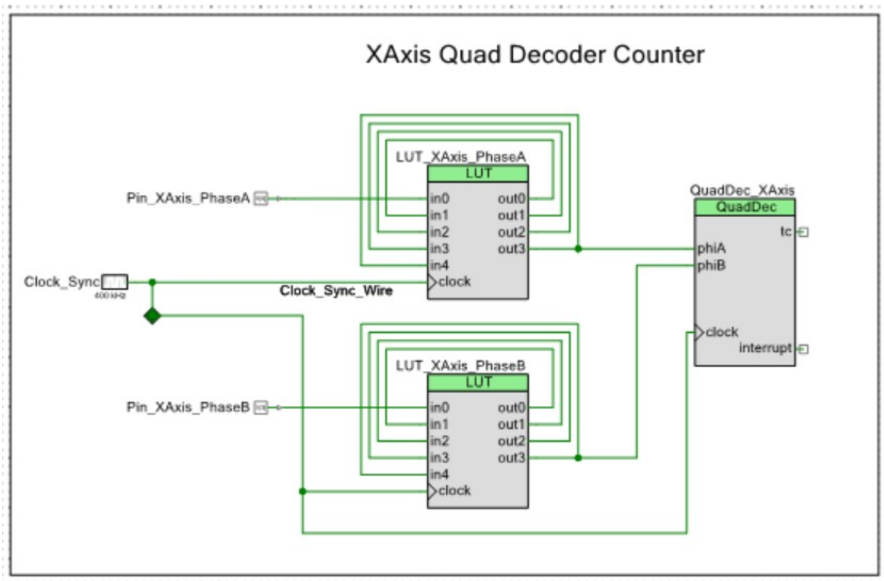


Figure 7: Y-Axis Quad Decoder Counter

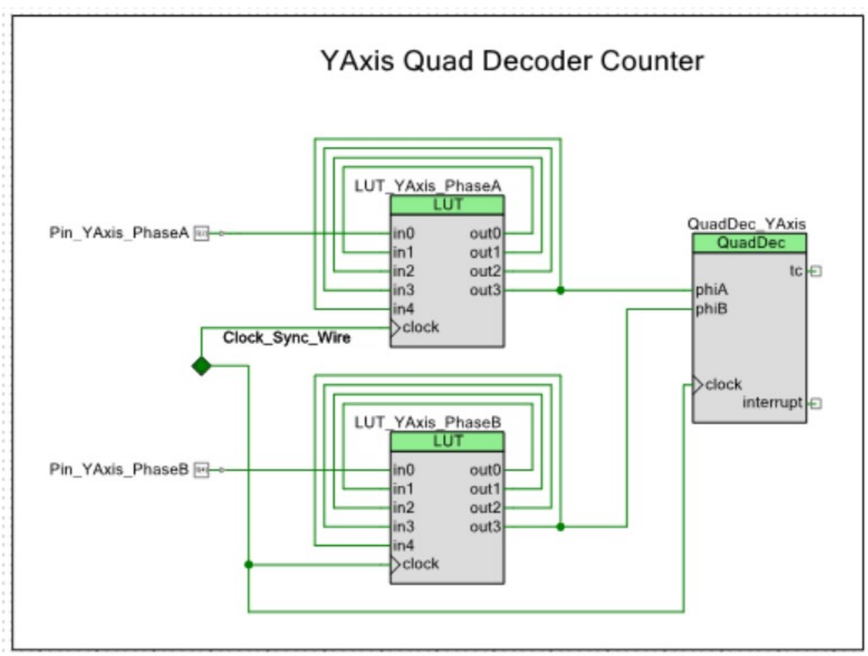
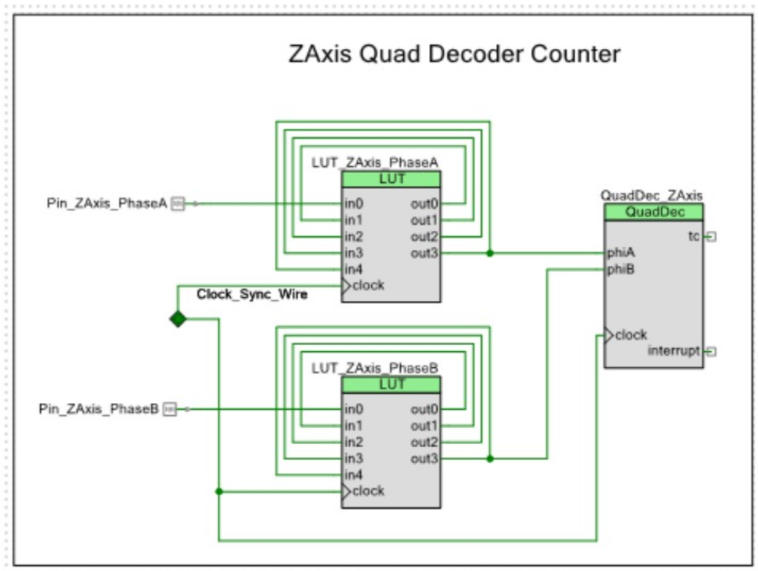


Figure 8: Z-Axis Quad Decoder Counter



Android Application Design:

The PSoC 6 BLE DRO Android App is used to interact with the PSoC 6 device which has the DRO project programmed into it. The application should be able to perform the following functionalities:

1. Scan for PSoC6 DRO Device and connect to the device.
2. Save user-provided full-scale length and full scale-count values that are required for calculation.
   Receive encoder counts from the PSoC 6 device and calculate the length measurement in each axis based on the full-scale length and full-scale count values.
3. Provide option to the user to set the workspace zero based on the user's set-up.

To develop this application, **Android Studio 4.0.1** is used which you can download from here. The requirements for the application is very similar to the Android application developed in the How to Create Android Apps to Interact with Cypress Bluetooth Low Energy. So, that would be the starting point for app development. If you have not covered the video tutorial series, please do that first.

When the App is opened, it scans for any devices with the same Service UUID and displays it to the user as shown in Figure 9. Once the user selects the device, the BLE connection is established and the notifications are enabled for all the 3 encoder characteristics. Inside the app, you can see the X axis, Y axis and Z axis values displaying, and it has **SET MACHINE PARAMETERS** and **WORKSPACE ZERO** as shown in Figure 10. The WORKSPACE ZERO button is used to send the reset command to the PSoC 6 device. The SET MACHINE PARAMETERS button enables the user to edit the full-scale length and full-scale count of the 3 axes that are needed for the calculation as shown in Figure 11. Shared Preferences are used to store these values and when the user presses the **SAVE** button, the values entered are checked if it is valid and then stored.

The length is calculated using the following equation –

$$length = \left(\frac{encoder\_count}{full\_scale\_count}\right) \times full\_scale\_length$$

where encoder_count is received from the PSoC 6 BLE device.

When the user quits the application, the resources are cleaned and the BLE connection is terminated.

Figure 9: PSoC6 DRO App

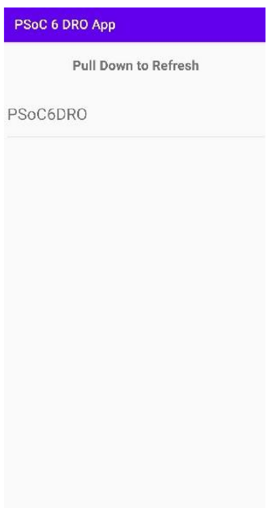


Figure 10: Count values in PSoC6 DRO App

Figure 11: FS Length and FS Count in PSoC6 DRO App

PSoC Components:

Table 1. PSoC Creator Components

Components	**Instance name**	**Hardware resource**	**Purpose**
BLE	BLE	1 BLE Component	BLE Communication
Quadrature Decoder	QuadDec_XAxis, QuadDec_YAxis, QuadDec_ZAxis,	3 TCPWM	32-bit Counter
Clock	Timer_Clock Clock_Sync	2 peripheral clock dividers	Generates programmable clock dividers for use with other Components that require clocks
Digital Output	Advertising_LED Disconnect_LED Connect_LED	3 GPIO pins	Drive blue, red and green LEDs
LUT	LUT_XAxis_PhaseA LUT_YAxis_PhaseA LUT_ZAxis_PhaseA LUT_XAxis_PhaseB	6 LUTs	Configured as a Glitch filter

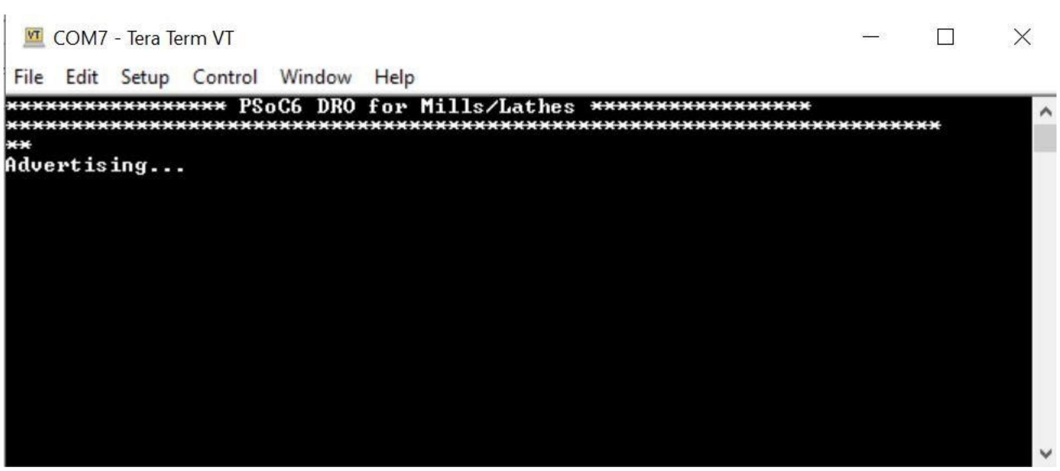
	LUT_YAxis_PhaseB LUT_ZAxis_PhaseB		
Digital Input	Pin_XAxis_PhaseA Pin_XAxis_PhaseB Pin_YAxis_PhaseA Pin_YAxis_PhaseB Pin_ZAxis_PhaseA Pin_ZAxis_PhaseB	6 GPIO pins	Configured the GPIO Component as digital input
UART	UART_DEB	1 SCB component	Print string messages on a terminal window
Timer	Timer_Tick	1 TCPWM	Generate an interrupt every 200milli-seconds
Interrupt	ISR_Tick	1 interrupt	Set the Timer component and priority for the interrupt

Operation:

1. Connect the CY8CKIT-062-BLE Pioneer Board to your PC using the provided USB cable through the USB connector (J10).
2. Connect your Linear encoders to the PSoC 6 Device digital input pins in the project.
3. Build the project and program the PSoC 6 MCU. For more information on device programming, see the Kit guide.
4. Open the serial terminal application, (this example uses TeraTerm) and set the baud rate to 115200 bps. Use the default settings for other serial port parameters. In the firmware, enable the below macro to see the DEBUG_PRINTF statements in terminal.

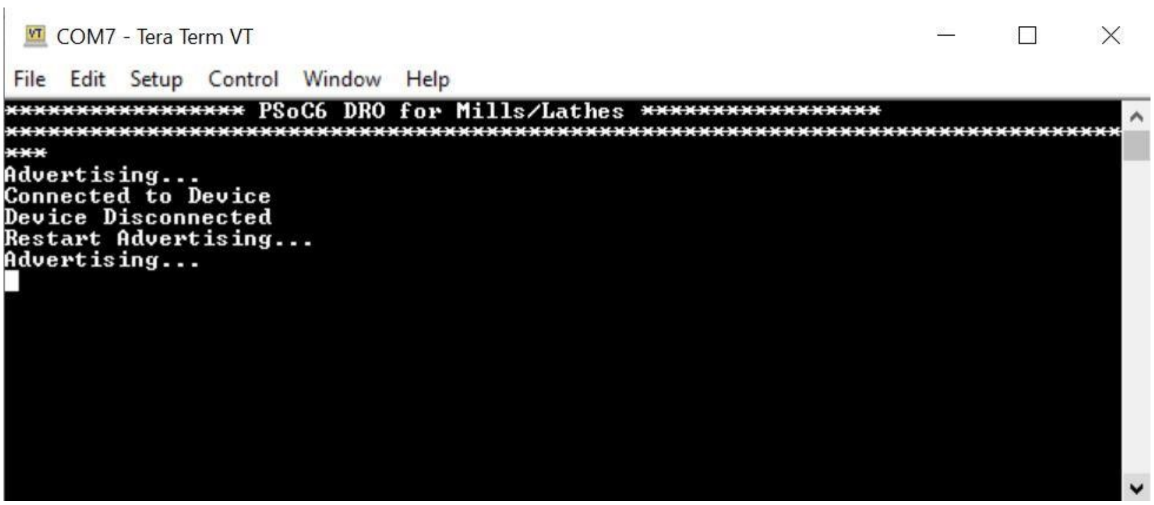
   #define DEBUG_UART_ENABLED ENABLED
5. Press the reset switch (SW1) on the kit. BLE will start advertising the device with the device name PSOC6DRO and this is indicated with a green LED on the Kit turning on. The status of the advertisement is also displayed on the UART terminal as shown below.

Figure 12: Serial Terminal showing the advertising state



6. In the Android APP, check for the PSOC6DRO name and connect to the device. The connection status by the device is indicated on the UART terminal and with a blue LED on the Kit.
7. The Timer_Tick component is configured with a 200ms period. For every 200ms, interrupt will be generated. Inside the ISR function, the notification enabled flags will be checked. If the notifications are enabled at the GATT Client device (BLE App), then for every 200ms notification data will be sent to the BLE App.
8. If the device gets disconnected, it restarts the advertisement. The disconnection status will be indicated with a red LED on the Kit. The status of disconnection is also displayed on the UART terminal as shown below.

Figure 10: Serial Terminal showing the disconnected state



9. In PSoC6 DRO APP, after the connection, you can see the X axis, Y axis, and Z axis values being displayed.